

**WEST**

Generate Collection

Print

L9: Entry 20 of 39

File: USPT

Aug 25, 1998

US-PAT-NO: 5799147

DOCUMENT-IDENTIFIER: US 5799147 A

TITLE: Computer recovery backup method

DATE-ISSUED: August 25, 1998

## INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Shannon; John P.	Yarmouth Port	MA	02675	

US-CL-CURRENT: 714/6; 714/15

## CLAIMS:

What is claimed is:

1. A computer file data backup method which comprises:

a) providing at least one client computer having a client disk image composed of a plurality of client data files, having a set of selected attributes, on a client computer storage means;

b) providing by the client computer a client disk map composed of the attributes of the client data files;

c) providing a separate server computer adapted to be connected periodically to said client computer and having server computer storage means only to receive, and to store said client disk image as a client logical disk image;

d) transmitting from said client computer by a client said client disk image in logical form into said server computer storage means to provide a stored client logical disk image;

e) updating periodically by the client in said client computer said client disk map by comparing said client disk map with the previous client disk map to identify any client data file with additions, modifications or deletions occurring since the last update of the previous client disk map to provide a revised client disk map;

f) transmitting from the client computer and only receiving and storing in said storage means of said server computer, the added or modified client data files and removing said deleted client data files to provide a revised updated client logical disk image;

g) acknowledging between said server computer and client computer the completing of the transmission;

h) storing in said server computer said revised updated client logical disk image; and

i) disconnecting the transmission link between said server computer and said client computer until the next client-initiated updating of said revised updated client logical disk image.

2. The method of claim 1 which includes manually initiating by a client user said updating, identifying and transmitting of said revised updated client logical disk image.

3. The method of claim 1 which includes preselecting, programming, automatically transmitting and storing of said client computer revised updated client logical disk image.
4. The method of claim 1 which includes transmitting between said client computer and said server computer by a public-switched telecommunications network system.
5. The method of claim 1 which includes positioning a plurality of client computers at separate geographical locations from said server computer, and providing transmission-connecting means between said client computer and said server computers.
6. The method of claim 5 wherein said client or server computers are selected from the group consisting of personal computers, laptop computers, and other storage means computers or combinations thereof.
7. The method of claim 1 wherein said client computer or said server computer includes optical or magnetic hard disks as said computer storage means.
8. The method of claim 1 which includes recovering by said client computer from said server computer storage means the last revised, updated client logical disk image from said server computer.
9. The method of claim 1 which includes auditing periodically by said server computer said stored, last revised, updated client logical disk image in said server computer.
10. The method of claim 1 which includes preventing the storing of the revised, updated client disk image, until all said modifications; additions and deletions have been completely transmitted and stored in said server computer storage means.
11. The method of claim 10 which includes, on interrupting of the transmitting step to said server computer, restarting automatically from the beginning of the transmitting step, and repeating the transmitting step to said server computer of said updated, revised client disk image.
12. A computer file data backup method which comprises:
  - a) providing a plurality of client computers at separate geographical locations on which backup protection is desired, each client computer having a client disk image composed of a plurality of client data files on a client computer storage means;
  - b) providing by each client computer a client disk map composed of attributes of the client data files;
  - c) providing a separate server computer adapted to be connected periodically to each of said client computers and having server computer storage means only to receive, and to store said client disk image of each client computer as a client logical disk image;
  - d) transmitting periodically from each client computer by a client said client disk image of said client computer in logical form into said server computer storage means to provide a stored client logical disk image;
  - e) updating periodically by the client, manually or automatically, in each of said client computers said client disk map by comparing said client disk map with the previous client disk map in said client computer to identify any client data file with additions, modifications or deletions occurring since the last update of the previous client disk map of said client computer to provide a revised client disk map;
  - f) transmitting as initiated by the client from each of the client computers by a telecommunication network system and receiving and storing in said storage means of said server computer, only the added or modified client data files and removing said deleted client data files to provide a revised updated client logical disk image;
  - g) acknowledging between said server computer and client computers the completing of the transmission to said client computer;
  - h) storing in the server computers the revised updated client logical disk image for said client computer; and

i) disconnecting the transmission link between said server computer and each of said client computers until the next client-initiated updating of said updated revised client disk image.

**WEST**

Generate Collection

Print

L9: Entry 25 of 39

File: USPT

Feb 17, 1998

US-PAT-NO: 5720026

DOCUMENT-IDENTIFIER: US 5720026 A

TITLE: Incremental backup system

DATE-ISSUED: February 17, 1998

## INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Uemura; Jose	Tokyo			JPX
Sakakura; Takashi	Tokyo			JPX

US-CL-CURRENT: 714/6; 707/204

## CLAIMS:

What is claimed is:

1. An incremental backup system, comprising:

a storage unit for storing data to be backed up, said storage unit being separated into a plurality of block units of a predetermined size and accessed according to the block units;

difference map information being stored in said storage unit and including for each block a latest backup generation number referencing a generation in which data has been updated in a block;

generation recording means for recording the latest backup generation number in said difference map information corresponding to a block in which data has been updated;

a backup unit for storing backup data; and

incremental backup means for inputting and storing backup data in said backup unit, the backup data including data stored in a block of said storage unit which is updated in a specified backup generation based on said difference map information, a position of the block in said storage unit, and a backup generation in which the block has been updated.

2. An incremental backup system as claimed in claim 1, further comprising full backup means for storing all of the data in said storage unit into said backup unit and initializing the backup generation numbers, wherein said incremental backup means updates the backup generation numbers after the backup data is stored.

3. An incremental backup system as claimed in claim 2, wherein said full backup means reserves an area for storing said difference map information in said storage unit and initializes said information.

4. An incremental backup system as claimed in claim 2, further comprising a controller for controlling said storage unit and said backup unit, said controller including said generation recording means, said incremental backup means, and said full backup means.

5. An incremental backup system as claimed in claim 1, wherein said incremental backup means comprises:

difference management means for outputting data, the data including data in a block of said storage unit updated in a specified backup generation based on said difference map information, a position of the block in said storage unit, and the backup generation in which the block has been updated; and

a control command for storing the data outputted by said difference management means in said backup unit as backup data.

6. An incremental backup system as claimed in claim 1, further comprising difference restoration means for inputting the backup data stored in said backup unit and restoring the input block data to said storage unit based on the position of the block in said storage unit recording and the backup generation in which the block has been updated in said difference map information.

7. An incremental backup system as claimed in claim 6, wherein said difference restoration means comprises:

a control command for inputting the backup data stored in said backup unit and outputting the inputted backup data; and

difference management means for inputting the backup data outputted by said control command and restoring the input block data to said storage unit based on the position of the block in said storage unit recording and the backup generation in which the block has been updated in said difference map information.

8. An incremental backup system as claimed in claim 6, wherein said incremental backup means stores check sum data corresponding to each block in said backup unit; and

wherein said difference restoration means uses the check sum data stored in said backup unit to check whether or not the block data stored in said backup unit is valid.

9. An incremental backup system as claimed in claim 1, further comprising number-of-update-blocks information for recording the number of data-update blocks for each backup generation; and

number-of-update-blocks recording means for adding the number of update blocks to said number-of-update-blocks information corresponding to the latest backup generation when the data is updated.

10. An incremental backup system as claimed in claim 9, further comprising a control command for inputting said number-of-update-blocks information and outputting the number of update blocks in said number-of-update-blocks information corresponding to the specified backup generation.

11. An incremental backup system as claimed in claim 1, wherein said storage unit is a virtual logical disk unit made up of physical disk units .

**WEST**

Generate Collection

Print

L9: Entry 14 of 39

File: USPT

May 25, 1999

US-PAT-NO: 5907672

DOCUMENT-IDENTIFIER: US 5907672 A

TITLE: System for backing up computer disk volumes with error remapping of flawed memory addresses

DATE-ISSUED: May 25, 1999

## INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Matze; John E. G.	Poway	CA		
Whiting; Douglas L.	Carlsbad	CA		

US-CL-CURRENT: 714/8; 711/162

## CLAIMS:

We claim:

1. A method for backing up data in a computer system from a primary storage means to a backup storage means on a sector-by-sector basis and restoring data in a computer system from said backup storage means to a restore storage means on a sector-by-sector basis, said method comprising the steps of:

reading a set of logically contiguous sectors from the primary storage means using a software call of the operating system that provides access to the files stored on said primary storage means, said call of said operating system performing any physical level remapping necessary to avoid previously detected physical flaws on said primary storage means,

writing said set of logically contiguous sectors to said backup storage means,

creating a partition on said restore storage means of a size at least as large as the size of said primary storage means,

reading a set of logically contiguous sectors from a location on said backup storage means,

writing said set of logically contiguous sectors to said partition of said restore storage means using a software call to the operating system that provides access to the files stored on said partition of said restore storage means, said call of said operating system performing any physical level remapping necessary to detect and avoid physical flaws on said restore storage means.

2. The method of claim 1, further including the steps of

writing on said backup storage means a sector directory table containing information sufficient to indicate the size of said primary storage means and the location of each logical sector written to said backup storage means,

reading said sector directory table from said backup storage means,

using said sector directory table to determine the sector numbers and locations of said logically contiguous blocks to be read.

3. The method of claim 2 wherein said primary storage means consists one or more disk drive partition(s) in said computer system, and wherein said operating system call to read said logically contiguous sectors performs the mapping necessary to locate said logically contiguous sectors on said disk drive partition(s).
4. The method of claim 2 wherein said partition created on said restore storage means is larger than the size of said original primary storage means.
5. The method of claim 2 wherein said partition created on said restore storage means spans multiple physical disk drives.
6. The method of claim 2 wherein unused sectors that do not contain file data are not read from said primary storage means and are not stored on said backup storage means, and wherein the absence of said unused sectors on the backup storage means is indicated in said sector directory table.
7. The method of any of claims 1-6 wherein deleted sectors that contain data from deleted files are not read from said primary storage means and are not stored on said backup storage means, and wherein the absence of said deleted sectors on the backup storage means is indicated in said sector directory table.
8. The method of any of claims 1-6 wherein an open file log is maintained of all files which are opened for write while the backup method is in process.
9. The method of claim 8 wherein said open file log is written to said backup storage means.
10. The method of any of claims 1-6 wherein said operating system allows multi-tasking, further including the step of:  
  
temporarily suspending execution of any tasks that attempt to write a set of sectors to said primary storage means until said set of sectors has been read from said primary storage means by the backup task in preparation for writing said sectors to said backup storage means.
11. The method of claim 10, further including the steps of:  
  
maintaining a cache of sectors read from said primary storage means to be written to said backup storage means,  
  
detecting an attempted write by a task to a set of sectors of said primary storage means which has not yet been backed up,  
  
operative when said sector cache is full, temporarily suspending executing of said task,  
  
operative when said sector cache is not full, reading said set of sectors and adding said set of sectors to said sector cache and then allowing said task to continue execution without suspension,  
  
checking said sector cache for the presence of any portion of said set of logically contiguous sectors to be read from said primary storage means,  
  
operative when no such portion is found in said sector cache, reading said set of logically contiguous sectors from said primary storage means,  
  
operative when such portion is found in said sector cache, reading said portion(s) of said set of logically contiguous sectors from said sector cache, and reading remaining portions not found in said sector cache from said primary storage means,  
  
whereby no said tasks attempting to write to said primary storage means will be suspended unless said sector cache is full.
12. The method of claim 11 wherein said portions of said set of logically contiguous set of sectors found in said sector cache are removed from said sector cache after said portions are read from said sector cache, whereby portions of said sector cache may be re-used in order to minimize the number of times that tasks are suspended.
13. The method of any of claims 1-6 wherein only sectors that have changed since the

last backup are written to the backup storage means.

14. The method of claim 13 wherein detection of changed sectors further includes the following steps:

computing a checksum (or similar type of function) on groups of sectors read from said primary storage means,

comparing said checksum with the corresponding checksum stored from the previous backup,

operative when the two checksums do not match,

writing said group of sectors to said backup storage means,

writing said checksum to said backup storage means,

operative when the two checksums do match,

setting the entry (or entries) in said sector directory table corresponding to said group of sectors to point to the corresponding group of sectors from said previous backup.

15. The method of claim 13 wherein detection of changed sectors further includes the following steps:

activating monitor software to detect all writes to said primary storage means,

maintaining a dirty sector table indicating which groups of sectors on said primary storage means have been modified,

using said dirty sector table to determine which groups of sectors have been changed,

operative when said dirty sector table indicates that said group of sectors to be backed up has been modified,

writing said group of sectors to said backup storage means,

operative when said dirty sector table indicates that said group of sectors to be backed up has not been modified,

setting the entry in said sector directory table corresponding to said group of sectors to point to the corresponding group of sectors from said previous backup,

saving said dirty sector table to an auxiliary storage means when said monitor software is deactivated at system shutdown.

16. The method of claim 15 wherein said auxiliary storage means is the same as said primary storage means.

17. The method of any of claims 15 further including the following steps:

operative when said monitor software is deactivated,

computing a checksum on the contents of said dirty sector table,

saving said checksum on said auxiliary storage means,

operative when said monitor software is activated,

performing a validity check on said contents of said dirty sector table using said checksum,

invalidating said checksum on said auxiliary storage means.

18. The method of claim 17 further including the steps of:

operative when said monitor software is deactivated,



saving an indicator of the time of said deactivation on said auxiliary storage means,  
operative when said monitor software is activated,

verifying that the operating system has not been active to allow writes to said primary storage means since the last time a valid dirty sector table was written to said auxiliary storage means,

operative when said verification fails, invalidating the contents of said dirty sector table.

19. The method of any of claims 18 wherein the failure of any checks on the validity of the contents of said dirty sector table results in all sectors being marked as having been modified, whereby a complete backup is performed.

20. The method of any of claims 1-6, further including the steps of:

creating a removable disk which contains all files necessary to boot said computer system into said operating system, including software drivers that allow access to said primary storage means and said backup storage means,

booting said computer system using said removable disk.

21. A method for backing up data in a computer system from a primary storage means to a backup storage means on a sector-by-sector basis and for providing file-by-file access to said data on said backup storage means, said method comprising the steps of:

reading a set of logically contiguous sectors from the primary storage means using a software call of the operating system that provides access to the files stored on said primary storage means, said call of said operating system performing any physical level remapping necessary to avoid previously detected physical flaws on said primary storage means,

writing said set of logically contiguous sectors to said backup storage means,

identifying a control set of logical sectors of said primary storage means, said control set including sectors required to mount said primary storage means for file access by said operating system or to traverse the directory structure of the files on said primary storage means,

re-ordering the sequence of writing said sets of logically contiguous sectors on said backup storage means in order to group sectors of said control set in closer physical proximity to one another on said backup storage means than would occur if said sequence were ordered strictly by logical sector number,

caching said control set of logical sectors from said backup storage means to allow fast random access to said control set,

creating a virtual disk partition of said operating system,

servicing logical sector read requests on said virtual disk partition,

operative when a sector of said read request is part of said control set, reading said sector from said control cache,

operative when a sector of said read request is not part of said control set, reading said sector from said backup storage means,

mounting said virtual disk partition as a disk volume of said operating system,

whereby files on said disk volume may be accessed using normal operating system calls and utilities.

22. The method of claim 21, further including the steps of:

writing on said backup storage means a sector directory table containing information sufficient to indicate the size of said primary storage means and the location of each logical sector written to said backup storage means,

reading said sector directory table from said backup storage means,

using said sector directory table to determine the locations of said sectors when servicing logical sector read requests.

23. The method of claim 22 wherein said primary storage means consists one or more disk drive partition(s) in said computer system, and wherein said operating system call to read said logically contiguous sectors performs the mapping necessary to locate said logically contiguous sectors on said disk drive partition(s).

24. The method of claim 22 wherein unused sectors that do not contain file data are not read from said primary storage means and are not stored on said backup storage means, and wherein the absence of said unused sectors on the backup storage means is indicated in said sector directory table.

25. The method of claim 24 wherein deleted sectors that contain data from deleted files are not read from said primary storage means and are not stored on said backup storage means, and wherein the absence of said deleted sectors on the backup storage means is indicated in said sector directory table.

26. The method of claim 21 wherein an open file log is maintained of all files which are opened for write while the backup method is in process.

27. The method of any of claims 21-26 wherein said open file log is written to said backup storage means.

28. The method of any of claims 21-26 wherein said operating system allows multi-tasking and further including the step of:

temporarily suspending execution of any tasks that attempt to write a set of sectors to said primary storage means until said set of sectors has been read from said primary storage means by the backup task in preparation for writing said sectors to said backup storage means.

29. The method of claim 28, further including the steps of:

maintaining a cache of sectors read from said primary storage means to be written to said backup storage means,

detecting an attempted write by a task to a set of sectors of said primary storage means which has not yet been backed up,

operative when said sector cache is full, temporarily suspending executing of said task,

operative when said sector cache is not full, reading said set of sectors and adding said set of sectors to said sector cache and then allowing said task to continue execution without suspension,

checking said sector cache for the presence of any portion of said set of logically contiguous sectors to be read from said primary storage means,

operative when no such portion is found in said sector cache, reading said set of logically contiguous sectors from said primary storage means,

operative when such portion is found in said sector cache, reading said portion(s) of said set of logically contiguous sectors from said sector cache, and reading remaining portions not found in said sector cache from said primary storage means,

whereby no said tasks attempting to write to said primary storage means will be suspended unless said sector cache is full.

30. The method of claim 29 wherein said portions of said set of logically contiguous set of sectors found in said sector cache are removed from said sector cache after said portions are read from said sector cache, whereby portions of said sector cache may be re-used in order to minimize the number of times that tasks are suspended.

31. The method of any of claims 21-26 wherein only sectors that have changed since the last backup are written to the backup storage means.

32. The method of claim 31 wherein detection of changed sectors further includes the following steps:

computing a checksum (or similar type of function) on groups of sectors read from said primary storage means,

comparing said checksum with the corresponding checksum stored from the previous backup,

operative when the two checksums do not match,

writing said group of sectors to said backup storage means,

writing said checksum to said backup storage means,

operative when the two checksums do match,

setting the entry (or entries) in said sector directory table corresponding to said group of sectors to point to the corresponding group of sectors from said previous backup.

33. The method of claim 31 wherein detection of changed sectors further includes the following steps:

activating monitor software to detect all writes to said primary storage means,

maintaining a dirty sector table indicating which groups of sectors on said primary storage means have been modified,

using said dirty sector table to determine which groups of sectors have been changed,

operative when said dirty sector table indicates that said group of sectors to be backed up has been modified,

writing said group of sectors to said backup storage means,

operative when said dirty sector table indicates that said group of sectors to be backed up has not been modified,

setting the entry in said sector directory table corresponding to said group of sectors to point to the corresponding group of sectors from said previous backup,

saving said dirty sector table to an auxiliary storage means when said monitor software is deactivated at system shutdown.

34. The method of claim 33 wherein said auxiliary storage means is the same as said primary storage means.

35. The method of claim 33 further including the following steps:

operative when said monitor software is deactivated,

computing a checksum on the contents of said dirty sector table,

saving said checksum on said auxiliary storage means,

operative when said monitor software is activated,

performing a validity check on said contents of said dirty sector table using said checksum,

invalidating said checksum on said auxiliary storage means.

36. The method of claim 35 further including the steps of:

operative when said monitor software is deactivated,

saving an indicator of the time of said deactivation on said auxiliary storage means,

operative when said monitor software is activated,

verifying that the operating system has not been active to allow writes to said primary storage means since the last time a valid dirty sector table was written to said auxiliary storage means,

operative when said verification fails, invalidating the contents of said dirty sector table.

37. The method of claim 36 wherein the failure of any checks on the validity of the contents of said dirty sector table results in all sectors being marked as having been modified, whereby a complete backup is performed.

38. The methods of claims 21-26 wherein said control set is identified by using a knowledge of the file and allocation format of said primary storage means under said operating system.

39. The method of any of claims 21-26 wherein said control set is identified without a complete knowledge of said file and allocation format of said primary storage means, using a pseudo-drive technique which includes the following steps:

creating a temporary virtual disk partition of said operating system,

servicing logical sector read requests on said temporary virtual disk partition by performing reads of the corresponding sectors of said primary storage means,

monitoring the set of logical sectors that are read from said temporary virtual disk partition and adding each sector read to said control set,

mounting said temporary virtual disk partition as a temporary disk volume of said operating system.

40. The method of claim 39, further including the step of using operating system calls to traverse the entire directory tree of said temporary disk volume.

41. The method of any of claims 21-26 wherein some knowledge of said file and allocation format is used to eliminate duplicate copies of structures in said primary storage means from said control set, whereby the size of said control set is minimized.

42. The method of any of claims 21-26 wherein writes to said disk volume are allowed by caching said writes to a temporary storage means.

43. The methods of claim 31 wherein said backup storage means can also be used to perform a sector-by-sector restore as in claim 1.

image.

5. The system of claim 1 wherein the means to transmit comprises a telecommunication network.
6. The system of claim 1 which includes a plurality of client computers at separate geographical locations from the server computer.
7. The system of claim 1 wherein said client or server computer storage means includes optical or magnetic disks.
8. The system of claim 1 which includes a means to recover, by said client computer from said server computer, the last revised, updated client logical disk image.
9. The system of claim 1 which includes a means to audit periodically, by said server, said stored, last revised, updated client logical disk image, in said server computer.
10. The system of claim 1 which includes a means to prevent the storing of the revised, updated client disk image until all additions, modifications, and deletions have been transmitted and stored in said server computer storage means.
11. The system of claim 1 which includes a means to restart on interruption automatically from the beginning of the transmission to said server computer of said updated, revised client disk image.
12. The system of claim 1 wherein said client or server computers comprise personal computers or lap top computers.
13. The system of claim 1 which includes a means to exclude specific client data files in the client disk map.